WHAT IS CLAIMED IS:

1. A computer-implemented method of searching a database for a prefix representing a destination address comprising:

loading two trees of tables, each tree of tables having a large table at a root branching to small tables; and

traversing the two tables of trees in parallel to find a match of an entry to the prefix.

2. The computer-implemented of claim 1 wherein an entry comprises:

a router pointer representing the destination address; and

a pointer to a next small table.

3. The computer-implemented method of 1 wherein the small tables comprise:

prefix match fields for indexed table entries;

a population count of pointers; and

hidden prefix entries that hold shorter prefix route entry pointers.

4. The computer-implemented method claim 1 further comprising reporting a non-match if the prefix does not match an entry.

5. The computer-implemented method of claim 1 wherein a first large table is a single 64k entry table that is indexed by bits 31:16 of an internet protocol (IP) address.

6. The computer-implemented method of claim 1 wherein a second large table is a single 256 entry table that is indexed by bits 31:24 of an internet protocol (IP) address.

7. The computer-implemented method of claim 5 wherein the small tables are dynamically allocated and comprise:

a tree with each node representing 4 bits of addresses covering an extension of 1-4 bits of a prefix entry from a previous tree.

8. The computer-implemented method of claim 6 wherein the small tables are dynamically allocated and comprise:

a tree with each node representing 4 bits of addresses covering an extension of 1-4 bits of a prefix entry from a previous tree.

9. A computer storage device storing a data structure for managing prefix representing internet protocol (IP) destination addresses, the data structure comprising:

two trees of tables, each tree of tables comprising:

a trie block, the trie block including a route pointer and a trie pointer;

a trie information structure, the trie information structure including masks and route entry pointers.

10. A computer-implemented method of searching a collection of data comprising:

searching a first table of trees and a second table of trees for a received search term, each of the trees of the first table and the second table containing a trie element and

a trie pointer, for a match of the search term with a trie element;

determining whether a trie pointer is non-null when the trie element matches the search term;

comparing a trie element in the tree of the first table containing the null pointer with a trie element in the tree of the second table containing the null pointer;

reporting a match if the search term matches the trie element in the first table of trees; and

reporting a match of the search term matches the trie element in the second table of trees.

11. The computer-implemented method of claim 10 wherein the search term is a destination address.

12. The computer-implemented method of claim 11 wherein the destination address is a prefix.

13. A computer program product, disposed on a computer readable medium, for searching a database for a prefix representing a destination address, the program comprising instructions for causing a computer to:

load two trees of tables, each tree of tables having a large table at a root branching to small tables; and

traverse the two tables of trees in parallel to find a match of an entry to the prefix.

14. The computer program of claim 13 wherein an entry comprises:

a router pointer representing the destination address; and

a pointer to a next small table.

15. The computer program of claim 13 wherein the small tables comprise:

prefix match fields for indexed table entries;

a population count of pointers; and

hidden prefix entries that hold shorter prefix route entry pointers.

16. The computer program claim 13 further comprising instructions for causing the computer to report a non-match if the prefix does not match an entry.

17. The computer program of claim 13 wherein a first large table is a single 64k entry table that is indexed by bits 31:16 of an internet protocol (IP) address.

18. The computer program of claim 13 wherein a second large table is a single 256 entry table that is indexed by bits 31:24 of an internet protocol (IP) address.

19. The computer program of claim 17 wherein the small tables are dynamically allocated and comprise a tree with each node representing 4 bits of addresses covering an extension of 1-4 bits of a prefix entry from a previous tree.

20. The computer program of claim 18 wherein the small tables are dynamically allocated and comprise a tree with each node representing 4 bits of addresses covering an extension of 1-4 bits of a prefix entry from a previous tree.

21. The computer-implemented method of claim 3 further comprising:

adding entries; and

deleting entries.

22. The computer-implemented method of claim 21 wherein deleting entries comprises:

removing corresponding trie entries;

decrementing the population counter;

determining an entry next longest prefix; and

inserting the next longest prefix in the trie.